Package 'MGMM'

August 26, 2020

Title Missingness Aware Gaussian Mixture Models

Date 2020-08-24

Version 0.3.1

Description Parameter estimation and classification for Gaussian Mixture Models (GMMs) in the presence of missing data. This package uses an expectation conditional maximization algorithm to obtain maximum likelihood estimates for all model parameters and maximum a posteriori classifications of the input vectors. For additional details, please see McCaw ZR, Julienne H, Aschard H. ``MGMM: an R package for fitting Gaussian Mixture Models on Incomplete Data." <doi:10.1101/2019.12.20.884551>.

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

LazyData true

LinkingTo Rcpp, RcppArmadillo

Imports cluster, methods, mvnfast, plyr, Rcpp (>= 1.0.3), stats

Suggests knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.1.1

NeedsCompilation yes

Author Zachary McCaw [aut, cre] (<https://orcid.org/0000-0002-2006-9828>)

Maintainer Zachary McCaw <zmccaw@alumni.harvard.edu>

Repository CRAN

Date/Publication 2020-08-26 11:50:02 UTC

R topics documented:

CalcWorkResp *Generate Working Response*

Description

Generate Working Response

Usage

```
CalcWorkResp(y, mean, cov, gamma)
```

Arguments

y	Vector with missing elements.
mean	Numeric mean.
cov	Numeric covariance.
gamma	Numeric vector of responsibilities.

Value

Numeric working response vector.

CalHar *Calinski-Harabaz Index*

Description

Calculates the Calinski-Harabaz index.

Usage

```
CalHar(data, assign, means)
```

Arguments

data	Observations.
assign	Assignments.
means	List of cluster means.

Value

Scalar index.

CalHar.within_cluster_disp *Within Cluster Dispersion*

Description

Within Cluster Dispersion

Usage

```
CalHar.within_cluster_disp(clust, mean)
```

Arguments

clust	Numeric data.matrix.
mean	Cluster mean.

Value

d x d within-cluster dispersion matrix.

ChooseK *Cluster Number Selection*

Description

Function to choose the number of clusters k. Examines cluster numbers between k0 and k1. For each cluster number, generates B bootstrap data sets, fits the Gaussian Mixture Model (fit.GMM), and calculates quality metrics (ClustQual). For each metric, determines the optimal cluster number k_opt, and the k_1SE, the smallest cluster number whose quality is within 1 SE of the optimum.

Usage

```
ChooseK(
  data,
  k0 = 2,
  k1 = NULL,
  boot = 100,
  init_means = NULL,
  fix_means = FALSE,
  init_covs = NULL,
  init_props = NULL,
  maxit = 10,
  eps = 1e-04,
  report = TRUE
)
```

Arguments

data	Numeric data matrix.
k0	Minimum number of clusters.
k1	Maximum number of clusters.
boot	Bootstrap replicates.
init_means	Optional list of initial mean vectors.
fix_means	Fix the means to their starting value? Must provide initial values.
init_covs	Optional list of initial covariance matrices.
init_props	Optional vector of initial cluster proportions.
maxit	Maximum number of EM iterations.
eps	Minimum acceptable increment in the EM objective.
report	Report bootstrap progress?

Value

List containing Choices, the recommended number of clusters according to each quality metric, and Results, the mean and standard error of the quality metrics at each cluster number evaluated.

See Also

See ClustQual for evaluating cluster quality, and fit.GMM for estimating the GMM with a specified cluster number.

Examples

```
set.seed(100)
mean_list <- list(c(2, 2), c(2, -2), c(-2, 2), c(-2, -2))
data <- rGMM(n = 500, d = 2, k = 4, means = mean_list)
choose_k <- ChooseK(data, k0 = 2, k1 = 6, boot = 10)
choose_k$Choices
```

ChooseK.bootstrap *Bootstrap Quality Metrics.*

Description

Bootstrap Quality Metrics.

Usage

```
ChooseK.bootstrap(
  boot,
  data,
  k,
  init_means,
  fix_means,
  init_covs,
  init_props,
  maxit,
  eps
)
```

Arguments

boot	Bootstrap replicates.
data	Numeric data matrix.
k	Number of clusters.
init_means	Optional list of initial mean vectors.
fix_means	Fix the means to their starting value? Must initialize.
init_covs	Optional list of initial covariance matrices.
init_props	Optional vector of initial cluster proportions.
maxit	Maximum number of EM iterations.
eps	Minimum acceptable increment in the EM objective.

Value

Numeric matrix of clustering metrics. Returns null if the models fails to fit the observed data.

ChooseK.iter *Attempt Model Fit and Return Quality Metrics.*

Description

Attempt Model Fit and Return Quality Metrics.

Usage

```
ChooseK.iter(data, k, init_means, fix_means, init_covs, init_props, maxit, eps)
```

Arguments

data	Numeric data matrix.
k	Number of clusters.
init_means	Optional list of initial mean vectors.
fix_means	Fix the means to their starting value? Must initialize.
init_covs	Optional list of initial covariance matrices.
init_props	Optional vector of initial cluster proportions.
maxit	Maximum number of EM iterations.
eps	Minimum acceptable increment in the EM objective.

Value

Numberic vector containing the 4 cluster quality metrics. Returns null if the model fails to fit.

ChooseK.recommend *Recommend Cluster Number based on Bootstrap Results.*

Description

Recommend Cluster Number based on Bootstrap Results.

Usage

```
ChooseK.recommend(results, metric, max_opt = FALSE)
```

Arguments

results	Data.frame of bootstrap results
metric	String, metric of interest.
max_opt	Is maximizing the metric optimal?

Value

Data.frame containing:

- ‘Metric‘ name.
- Optimal cluster number ‘k_opt‘ and metric value ‘Metric_opt‘.
- 1SE cluster number ‘k_1se‘ and metric value ‘Metric_1se‘.

ChooseK.summarize *Summarize Results of Quality Metric Bootstrap.*

Description

Summarize Results of Quality Metric Bootstrap.

Usage

```
ChooseK.summarize(k, boot_metrics, report)
```

Arguments

k	Clusters.
boot_metrics	Bootstrapped quality metrics.
report	Report bootstrap results?

Value

Either a data.table reporting the means and standard errors of quality metrics, or NULL if too few fits succeded to calculate the standard errors.

ClustQual *Cluster Quality*

Description

Evaluates cluster quality. Returns the following metrics:

- BIC: Bayesian Information Criterion, lower value indicates better clustering quality.
- CHI: Calinski-Harabaz Index, higher value indicates better clustering quality.
- DBI: Davies-Bouldin, lower value indicates better clustering quality.
- SIL: Silhouette Width, higher value indicates better clustering quality.

Usage

```
ClustQual(fit)
```

Arguments

fit	Object of class mix.

Value

List containing the cluster quality metrics.

See Also

See ChooseK for using quality metrics to choose the cluster number.

Examples

```
## Not run:
set.seed(100)
# Data generation
mean_list = list(
c(2, 2, 2),
c(-2, 2, 2),
c(2, -2, 2),
c(2, 2, -2)
)

data <- rGMM(n = 500, d = 3, k = 4, means = mean_list)
fit <- fit.GMM(data, k = 4)

# Clustering quality
cluster_qual <- ClustQual(fit)

## End(Not run)
```

ClustQual.partition_by_clust

Partition Data by Cluster Assignment.

Description

Partition Data by Cluster Assignment.

Usage

```
ClustQual.partition_by_clust(data, assign)
```

Arguments

data	Numeric data.matrix.
assign	Cluster assignments.

Value

List of numeric data.matrices, separated by cluster assignment.

DavBou *Davies-Bouldin Index*

Description

Calculates the Davies-Bouldin index.

Usage

```
DavBou(data, assign, means)
```

Arguments

data	Observations
assign	Assignments
means	List of cluster means

Value

Scalar index.

DavBou.clust_diameter *Mean Cluster Diameter*

Description

Mean Cluster Diameter

Usage

```
DavBou.clust_diameter(clust, mean)
```

Arguments

clust	Numeric data.matrix.
mean	Cluster mean.

Value

Scalar mean diameter.

eigSym *Eigenvalues of Symmetric Matrix.*

Description

Calculates the eigenvalues of a symmetric matrix.

Usage

```
eigSym(A)
```

Arguments

A	symmetric matrix.

Value

Numeric vector.

ExpResidOP *Expected Residual Outer Product*

Description

Calculates the expected residual outer product.

Usage

```
ExpResidOP(data_incomp, new_mean, old_mean, old_cov, gamma = NULL)
```

Arguments

data_incomp	Data for observations with missingness.
new_mean	New mean.
old_mean	Initial mean.
old_cov	Initial covariance.
gamma	Responsibilities.

Value

Numeric matrix, the responsibility-weighted, cumulative, expected residual outer product.

`fit.GMM` *Estimate Multivariate Normal Mixture*

Description

Given an $n \times d$ matrix of random vectors, estimates the parameters of a Gaussian Mixture Model (GMM). Accommodates arbitrary patterns of missingness at random (MAR) in the input vectors.

Usage

```
fit.GMM(
  data,
  k = 1,
  init_means = NULL,
  fix_means = FALSE,
  init_covs = NULL,
  init_props = NULL,
  maxit = 100,
  eps = 1e-06,
  report = TRUE
)
```

Arguments

`data`	Numeric data matrix.
`k`	Number of mixture components. Defaults to 1.
`init_means`	Optional list of initial mean vectors.
`fix_means`	Fix the means to their starting value? Must provide initial values.
`init_covs`	Optional list of initial covariance matrices.
`init_props`	Optional vector of initial cluster proportions.
`maxit`	Maximum number of EM iterations.
`eps`	Minimum acceptable increment in the EM objective.
`report`	Report fitting progress?

Details

Initial values for the cluster means, covariances, and proportions are specified using `M0`, `S0`, and `pi0`, respectively. If the data contains complete observations, i.e. observations with no missing elements, then `fit.GMM` will attempt to initialize these parameters internally using K-means. If the data contains no complete observations, then initial values are required for `M0`, `S0`, and `pi0`.

Value

For a single component model $k = 1$, a list is returned, containing the estimated mean, covariance, and final EM objective. For a multi-component model $k > 1$, an object of class `mix`, containing the estimated means, covariances, cluster proportions, cluster responsibilities, and observation assignments.

See Also

See rGMM for data generation, and ChooseK for selecting the number of clusters.

Examples

```
# Single component without missingness
# Bivariate normal observations
sigma <- matrix(c(1, 0.5, 0.5, 1), nrow = 2)
data <- rGMM(n = 1e3, d = 2, k = 1, means = c(2, 2), covs = sigma)
fit <- fit.GMM(data, k = 1)

# Single component with missingness
# Trivariate normal observations
mean_list <- list(c(-2, -2, -2), c(2, 2, 2))
sigma <- matrix(c(1, 0.5, 0.5, 0.5, 1, 0.5, 0.5, 0.5, 1), nrow = 3)
data <- rGMM(n = 1e3, d = 3, k = 2, means = mean_list, covs = sigma)
fit <- fit.GMM(data, k = 2)

# Two components without missingness
# Trivariate normal observations
mean_list <- list(c(-2, -2, -2), c(2, 2, 2))
sigma <- matrix(c(1, 0.5, 0.5, 0.5, 1, 0.5, 0.5, 0.5, 1), nrow = 3)
data <- rGMM(n = 1e3, d = 3, k = 2, means = mean_list, covs = sigma)
fit <- fit.GMM(data, k = 2)

# Four components with missingness
# Bivariate normal observations
# Note: Fitting is slow.
mean_list <- list(c(2, 2), c(2, -2), c(-2, 2), c(-2, -2))
sigma <- 0.5 * diag(2)
data <- rGMM(
n = 1000,
d = 2,
k = 4,
pi = c(0.35, 0.15, 0.15, 0.35),
m = 0.1,
means = mean_list,
covs = sigma)
fit <- fit.GMM(data, k = 4)
```

---

fit.mix *Fit Multivariate Mixture Distribution*

Description

Given a matrix of random vectors, estimates the parameters for a mixture of multivariate normal distributions. Accommodates arbitrary patterns of missingness, provided the elements are missing at random (MAR).

Usage

```
fit.mix(
  data,
  k = 2,
  init_means = NULL,
  fix_means = FALSE,
  init_covs = NULL,
  init_props = NULL,
  maxit = 100,
  eps = 1e-06,
  report = FALSE
)
```

Arguments

data	Numeric data matrix.
k	Number of mixture components. Defaults to 2.
init_means	Optional list of initial mean vectors.
fix_means	Fix means to their starting values? Must initialize.
init_covs	Optional list of initial covariance matrices.
init_props	Optional vector of initial cluster proportions.
maxit	Maximum number of EM iterations.
eps	Minimum acceptable increment in the EM objective.
report	Report fitting progress?

Value

Object of class `mix` containing the estimated

---

fit.mix.miss.impute *Imputation for Mixutre of MVNs with Missingness.*

Description

Imputation for Mixutre of MVNs with Missingness.

Usage

```
fit.mix.miss.impute(split_data, theta)
```

Arguments

split_data	Data partitioned by missingness.
theta	List containing the current ‘means‘, ‘covs‘, ‘pi‘, and ‘gamma‘.

Value

Data.matrix, in the same order as the original data, with missing values imputed to their expectations.

`fit.mix.miss.init` *Parameter Initialization for Mixture of Multivariate Normals.*

Description

Parameter Initialization for Mixture of Multivariate Normals.

Usage

```
fit.mix.miss.init(split_data, k, init_means, init_covs, init_props)
```

Arguments

`split_data`	Data partitioned by missingness.
`k`	Number of mixture components.
`init_means`	Optional list of initial mean vectors.
`init_covs`	Optional list of initial covariance matrices.
`init_props`	Optional vector of initial cluster proportions.

`fit.mix.miss.update` *Parameter Update for Mixutre of MVNs with Missingness.*

Description

Parameter Update for Mixutre of MVNs with Missingness.

Usage

```
fit.mix.miss.update(split_data, theta, fix_means)
```

Arguments

`split_data`	Data partitioned by missingness.
`theta`	List containing the current ‘means‘, ‘covs‘, ‘pi‘, and ‘gamma‘.
`fix_means`	Fix the mean to its starting value? Must initialize.

Value

List containing:

- The updated ‘mean‘, ‘cov‘, ‘pi‘, and ‘gamma‘.
- The initial ‘old_obj‘ and final ‘new_obj‘ EM objective.
- The increase in the EM objective ‘delta‘.

fit.mix.miss.update.means *Mean Update for Mixture of MVNs with Missingness.*

Description

Mean Update for Mixture of MVNs with Missingness.

Usage

```
fit.mix.miss.update.means(split_data, means, covs, gamma)
```

Arguments

split_data	Data partitioned by missingness.
means	List of component means.
covs	List of component covariances.
gamma	List of component responsibilities.

Value

List containing the updated component means.

fit.mvn *Fit Multivariate Normal Distribution*

Description

Given a matrix of n x d-dimensional random vectors, possibly containing missing elements, estimates the mean and covariance of the best fitting multivariate normal distribution.

Usage

```
fit.mvn(
  data,
  init_mean = NULL,
  fix_mean = FALSE,
  init_cov = NULL,
  maxit = 100,
  eps = 1e-06,
  report = TRUE
)
```

Arguments

data	Numeric data matrix.
init_mean	Optional initial mean vector.
fix_mean	Fix the mean to its starting value? Must initialize.
init_cov	Optional initial covariance matrix.
maxit	Maximum number of EM iterations.
eps	Minimum acceptable increment in the EM objective.
report	Report fitting progress?

Value

List containing the final 'Mean', 'Covariance', and EM 'Objective'. If missing data are present, a 'Completed' data matrix is also returned.

---

fit.mvn.miss *Estimation for a Single Component Multivariate Normal with Missingness.*

Description

Estimation for a Single Component Multivariate Normal with Missingness.

Usage

```
fit.mvn.miss(data, init_mean, fix_mean, init_cov, maxit, eps, report)
```

Arguments

data	Numeric data matrix.
init_mean	Optional initial mean vector.
fix_mean	Fix the means to their starting value? Must initialize.
init_cov	Optional initial covariance matrix.
maxit	Maximum number of EM iterations.
eps	Minimum acceptable increment in the EM objective.
report	Report fitting progress?

Value

List containing the final 'Mean', 'Covariance', EM 'Objective' and a 'Completed' data matrix.

`fit.mvn.miss.impute` *Imputation for a Single Component Multivariate Normal with Missingness.*

Description

Imputation for a Single Component Multivariate Normal with Missingness.

Usage

```
fit.mvn.miss.impute(split_data, theta)
```

Arguments

`split_data`	Data partitioned by missingness.
`theta`	List containing the ‘mean‘ and ‘cov‘.

Value

Data.matrix, in the same order as the original data, with missing values imputed to their expectations.

`fit.mvn.miss.init` *Parameter Initialization for Single Component Multivariate Normal with Missingness.*

Description

Parameter Initialization for Single Component Multivariate Normal with Missingness.

Usage

```
fit.mvn.miss.init(data_comp, init_mean, init_cov)
```

Arguments

`data_comp`	Complete cases.
`init_mean`	Optional initial mean vector.
`init_cov`	Optional initial covariance matrix.

Value

List containing the initialized ‘mean‘ and ‘cov‘.

fit.mvn.miss.update — *Parameter Update for Single Component Multivariate Normal with Missingness.*

Description

Parameter Update for Single Component Multivariate Normal with Missingness.

Usage

```
fit.mvn.miss.update(split_data, theta, fix_mean)
```

Arguments

split_data	Data partitioned by missingness.
theta	List containing the current ‘mean‘ and ‘cov‘.
fix_mean	Fix the mean to its starting value? Must initialize.

Value

List containing:

- The updated ‘mean‘ and ‘cov‘.
- The initial ‘old_obj‘ and final ‘new_obj‘ EM objective.
- The increase in the EM objective ‘delta‘.

fit.mvn.no_miss — *Estimation for a Single Component Multivariate Normal with No Missingness.*

Description

Estimation for a Single Component Multivariate Normal with No Missingness.

Usage

```
fit.mvn.no_miss(data, init_mean, fix_mean)
```

Arguments

data	Numeric data matrix.
init_mean	Optional initial mean vector.
fix_mean	Fix the means to their starting value? Must initialize

Value

List containing the new ‘Mean‘, ‘Covariance‘, and log-likelihood ‘Objective‘.

matCov *Covariance*

Description

Calculates the correlation between two matrices.

Usage

```
matCov(A, B, corMat = FALSE)
```

Arguments

A	NxP matrix.
B	NxQ matrix.
corMat	Return correlation matrix? If false, returns a covariance matrix.

Value

Numeric matrix.

matDet *Matrix Determinant*

Description

Calculates the determinant of A.

Usage

```
matDet(A, logDet = FALSE)
```

Arguments

A	Numeric matrix.
logDet	Return the logarithm of the determinant?

Value

Scalar.

matInv *Matrix Inverse*

Description

Calcualtes A^{-1}.

Usage

```
matInv(A)
```

Arguments

A	Numeric matrix.

Value

Numeric matrix.

matIP *Matrix Inner Product*

Description

Calculates the product $A'B$.

Usage

```
matIP(A, B)
```

Arguments

A	Numeric matrix.
B	Numeric matrix.

Value

Numeric matrix.

matOP *Matrix Outer Product*

Description

Calculates the outer product AB'.

Usage

```
matOP(A, B)
```

Arguments

A	Numeric matrix.
B	Numeric matrix.

Value

Numeric matrix.

matQF *Quadratic Form*

Description

Calculates the quadratic form $X'AX$.

Usage

```
matQF(X, A)
```

Arguments

X	Numeric matrix.
A	Numeric matrix.

Value

Numeric matrix.

Maximization *Maximization Iteration.*

Description

Maximization Iteration.

Usage

```
Maximization(theta0, Update, maxit, eps, report)
```

Arguments

theta0	List containing the initial values of ‘mean‘ and ‘cov‘.
Update	Function to iteratively update theta0.
maxit	Maximum number of EM iterations.
eps	Minimum acceptable increment in the EM objective.
report	Report fitting progress?

Value

List containing the updated ‘mean‘ and ‘cov‘, the final EM objective ‘new_obj‘.

MGMM *MGMM: Missingness Aware Gaussian Mixture Models*

Description

Parameter estimation and classification via Gaussian Mixture Models, allowing for missingness in the input vectors. See `fit.GMM` for estimating the GMM, and `ChooseK` for selecting the number of clusters. See `rGMM` for simulating data from a GMM.

Author(s)

Zachary R. McCaw

mix-class *Mixture Model Class*

Description

Defines the class returned by the fitting functions.

Slots

`Assignments` Maximum a posteriori assignment.

`Completed` Completed data, with missing values replaced by posterior expectations.

`Components` Components.

`Covariances` Fitted cluster covariances.

`Density` Component density at observation.

`Means` Fitted cluster means.

`Objective` Final value of the EM objective.

`Proportions` Fitted cluster proportions.

`Responsibilities` Posterior membership probabilities.

MixClusterAssign *Cluster Assignment for Mixutre of MVNs with Missingness.*

Description

Cluster Assignment for Mixutre of MVNs with Missingness.

Usage

```
MixClusterAssign(split_data, theta)
```

Arguments

`split_data`	Data partitioned by missingness.
`theta`	List containing the current ‘means‘, ‘covs‘, ‘pi‘, and ‘gamma‘.

Value

List containing:

- Matrix of cluster ‘Assignments‘.
- Matrix of ‘Density‘ evaluations.
- Matrix of cluster ‘Responsibilities‘.

`MixClusterSizes` *Cluster Sizes for a Mixutre of MVNs.*

Description

Cluster Sizes for a Mixutre of MVNs.

Usage

```
MixClusterSizes(split_data, gamma)
```

Arguments

`split_data`	Data partitioned by missingness.
`gamma`	List cof component responsibilities.

`MixEMObj` *EM Objective for a Mixture of MVNs.*

Description

EM Objective for a Mixture of MVNs.

Usage

```
MixEMObj(cluster_sizes, pi, covs, resid_ops)
```

Arguments

`cluster_sizes`	Cluster sizes.
`pi`	Cluster proportions
`covs`	List of component covariances.
`resid_ops`	List of residual outer products.

MixResidOP *Expected Residual Outer Product for a Mixutre of MVNs.*

Description

Expected Residual Outer Product for a Mixutre of MVNs.

Usage

```
MixResidOP(split_data, new_means, old_means, covs, gamma)
```

Arguments

split_data	Data partitioned by missingness.
new_means	List of updated means.
old_means	List of previous means.
covs	List of component covariances.
gamma	List cof component responsibilities.

Value

List of k expected residual outer products.

MMP *Matrix Matrix Product*

Description

Calculates the product AB.

Usage

```
MMP(A, B)
```

Arguments

A	Numeric matrix.
B	Numeric matrix.

Value

Numeric matrix.

PartitionData *Paratition a Data.frame by Missingness cases.*

Description

Paratition a Data.frame by Missingness cases.

Usage

```
PartitionData(data)
```

Arguments

data	Data.frame.

Value

List containing:

- The original row and column names: ‘orig_row_names‘, ‘orig_col_names‘.
- The original row and column numbers: ‘n_row‘ and ‘n_col‘.
- The complete cases ‘data_comp‘.
- The incomplete cases ‘data_incomp‘.
- The empty cases ‘data_empty‘.
- Counts of complete ‘n0‘, incomplete ‘n1‘, and empty ‘n2‘ cases.
- Initial order of the observations ‘init_order‘.

print.mix *Print for Fitted Mixture Model*

Description

Print method for objects of class mix.

Usage

```
## S3 method for class 'mix'
print(x, ...)
```

Arguments

x	A mix object.
...	Unused.

`Responsibility` *Responsibilities*

Description

Calculates the posterior probability of cluster membership given the observed data.

Usage

```
Responsibility(split_data, means, covs, pi)
```

Arguments

`split_data`	Data partitioned by missingness.
`means`	List of mean vectors.
`covs`	List of covariances matrices.
`pi`	Vector of cluster proportions.

Value

List containing:

- k Number of mixture components.
- Density evaluations ‘dens_eval0‘ and responsibilities ‘gamma0‘ for complete cases.
- Density evaluations ‘dens_eval1‘ and responsibilities ‘gamma1‘ for incomplete cases.

`Responsibility.eval_dens_incomp`
Evaluate the Density of an Incomplete Observations

Description

Evaluate the Density of an Incomplete Observations

Usage

```
Responsibility.eval_dens_incomp(y, means, covs, pi)
```

Arguments

`y`	Vector with missing elements.
`means`	List of mean vectors.
`covs`	List of covariances matrices.
`pi`	Vector of cluster proportions.

Value

Numeric density of observed elements.

---

rGMM *Data Generation from Multivariate Normal Mixture Models*

Description

Generates an $n \times d$ matrix of multivariate normal random vectors with observations as rows. If $k = 1$, all observations belong to the same cluster. If $k > 1$ the observations are generated via two-step precoedure. First, the cluster membership is drawn from a multinomial distribution, with mixture proportions specified by `pi`. Conditional on cluster membership, the observation is drawn from a multivariate normal distribution, with cluster-specific mean and covariance. The cluster means are provided using `M`, and the cluster covariance matrices are provided using `covs`. If $m > 0$, missingness is introduced, completely at random, by setting that proportion of elements in the data matrix to `NA`.

Usage

```
rGMM(n, d = 2, k = 1, pi = NULL, miss = 0, means = NULL, covs = NULL)
```

Arguments

`n`	Observations (rows).
`d`	Observation dimension (columns).
`k`	Number of mixture components. Defaults to 1.
`pi`	Mixture proportions. If omitted, components are assumed equi-probable.
`miss`	Proportion of elements missing, $miss \in [0, 1)$.
`means`	Either a prototype mean vector, or a list of mean vectors. Defaults to the zero vector.
`covs`	Either a prototype covariance matrix, or a list of covariance matrices. Defaults to the identity matrix.

Value

Numeric matrix with observations as rows. Row numbers specify the true cluster assignments.

See Also

For estimation, see `fit.GMM`.

Examples

```
set.seed(100)
# Single component without missingness
# Bivariate normal observations
cov <- matrix(c(1, 0.5, 0.5, 1), nrow = 2)
data <- rGMM(n = 1e3, d = 2, k = 1, means = c(2, 2), covs = cov)

# Single component with missingness
# Trivariate normal observations
mean_list <- list(c(-2, -2, -2), c(2, 2, 2))
cov <- matrix(c(1, 0.5, 0.5, 0.5, 1, 0.5, 0.5, 0.5, 1), nrow = 3)
data <- rGMM(n = 1e3, d = 3, k = 2, means = mean_list, covs = cov)

# Two components without missingness
# Trivariate normal observations
mean_list <- list(c(-2, -2, -2), c(2, 2, 2))
cov <- matrix(c(1, 0.5, 0.5, 0.5, 1, 0.5, 0.5, 0.5, 1), nrow = 3)
data <- rGMM(n = 1e3, d = 3, k = 2, means = mean_list, covs = cov)

# Four components with missingness
# Bivariate normal observations
mean_list <- list(c(2, 2), c(2, -2), c(-2, 2), c(-2, -2))
cov <- 0.5 * diag(2)
data <- rGMM(
n = 1000,
d = 2,
k = 4,
pi = c(0.35, 0.15, 0.15, 0.35),
miss = 0.1,
means = mean_list,
covs = cov)
```

`SchurC` *Schur complement*

Description

Calculates the efficient information $I_{bb} - I_{ba} I_{aa}^{-1} I_{ab}$.

Usage

```
SchurC(Ibb, Iaa, Iba)
```

Arguments

`Ibb`	Information of target parameter
`Iaa`	Information of nuisance parameter
`Iba`	Cross information between target and nuisance parameters

Value

Numeric matrix.

`show,mix-method` *Show for Fitted Mixture Models*

Description

Show for Fitted Mixture Models

Usage

```
## S4 method for signature 'mix'
show(object)
```

Arguments

`object`	A `mix` object.

`tr` *Matrix Trace*

Description

Calculates the trace of a matrix A.

Usage

```
tr(A)
```

Arguments

`A`	Numeric matrix.

Value

Scalar.

`WorkResp` *Working Response Vectors*

Description

Calculate the working response vectors.

Usage

```
WorkResp(data_incomp, mean, cov, gamma = NULL)
```

Arguments

`data_incomp`	Incomplete observations.
`mean`	Numeric mean.
`cov`	Numeric covariance.
`gamma`	Numeric vector of responsibilities.

Value

Numeric vector, the responsibility-weighted cumulative working response vector.

Index